

# Journal of Electronic Imaging

JElectronicImaging.org

## **Operationally optimal vertex-based shape coding with arbitrary direction edge encoding structures**

Zhongyuan Lai  
Junhuan Zhu  
Jiebo Luo

# Operationally optimal vertex-based shape coding with arbitrary direction edge encoding structures

Zhongyuan Lai,<sup>a,\*</sup> Junhuan Zhu,<sup>b</sup> and Jiebo Luo<sup>b</sup>

<sup>a</sup>Jiangnan University, Institute for Interdisciplinary Research, Wuhan Economic and Technological Development Zone, Wuhan 430056, China

<sup>b</sup>University of Rochester, Department of Computer Science, Rochester, New York 14627, United States

**Abstract.** The intention of shape coding in the MPEG-4 is to improve the coding efficiency as well as to facilitate the object-oriented applications, such as shape-based object recognition and retrieval. These require both efficient shape compression and effective shape description. Although these two issues have been intensively investigated in data compression and pattern recognition fields separately, it remains an open problem when both objectives need to be considered together. To achieve high coding gain, the operational rate-distortion optimal framework can be applied, but the direction restriction of the traditional eight-direction edge encoding structure reduces its compression efficiency and description effectiveness. We present two arbitrary direction edge encoding structures to relax this direction restriction. They consist of a sector number, a short component, and a long component, which represent both the direction and the magnitude information of an encoding edge. Experiments on both shape coding and hand gesture recognition validate that our structures can reduce a large number of encoding vertices and save up to 48.9% bits. Besides, the object contours are effectively described and suitable for the object-oriented applications. © The Authors. Published by SPIE under a Creative Commons Attribution 3.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: 10.1117/1.JEI.23.4.043009]

Keywords: operational rate-distortion theory; edge encoding structure; shape coding; hand gesture recognition.

Paper 14061 received Feb. 4, 2014; revised manuscript received Apr. 30, 2014; accepted for publication Jun. 18, 2014; published online Jul. 15, 2014.

## 1 Introduction

To facilitate the applications of object-oriented storage, retrieval, editing, and interaction, modern multimedia communications require that video content has to be easily accessible on an object basis. The MPEG-4 video compression standard provides such functionalities by describing video objects not only by texture but also by shape. Because of severe mobile environments and massive image and video retrieval demands, a good shape coding scheme has to provide an efficient compression as well as effective description.<sup>1</sup> These two requirements have been extensively investigated in data compression<sup>2-4</sup> and pattern recognition fields<sup>5-7</sup> separately in two recent decades. However, jointly considering both objectives in one framework remains an open problem.

It is well known that the vertex-based shape representation can handle both shape coding and shape description in a natural way. As a result, this representation can be directly applied to the contour-based object-oriented applications. In general, there are two main vertex-based shape coding frameworks. One is the iterative refinement framework.<sup>2-4</sup> It treats the vertex selection and encoding separately; therefore, it loses the rate-distortion (RD) optimality. The other is the operational rate-distortion (ORD) optimal framework.<sup>1,8,9</sup> It jointly considers the vertex selection and encoding as a shortest path problem in a directed acyclic graph (DAG); therefore, it can guarantee the optimality in the RD sense.

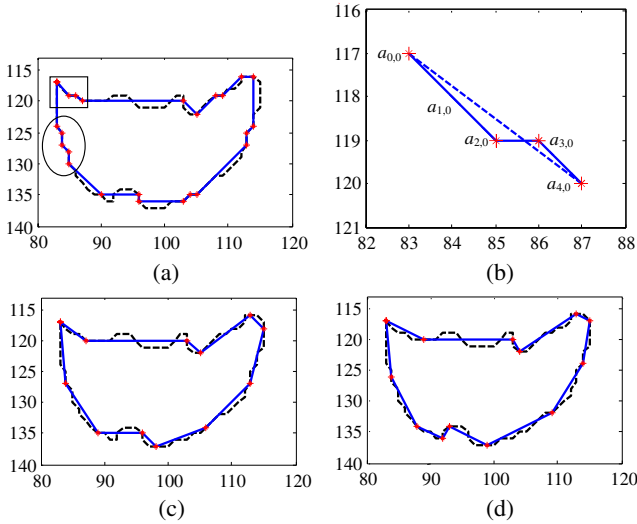
The main limitation of the ORD optimal framework, however, is that the optimality is contingent on the chosen parameters. Thus, the performance enhancements to this

framework are to relax the constraints of these parameters. Many relaxations used in the ORD optimal framework have been focused on the admissible vertex set,<sup>1,9,10</sup> the sliding window strategy,<sup>1,10,11</sup> the edge distortion measurement,<sup>1,12-16</sup> and the code table,<sup>17-19</sup> but few have been concentrated on the edge encoding structure. Their problem to be addressed is to find an optimal polygon that can be encoded with the lowest bit rate for a given admissible distortion, where the optimality is contingent on the given eight-direction edge encoding structure.<sup>20</sup> Since the location of the current vertex is strongly correlated with the location of the previous one, it is often assumed that the vertices are encoded differentially. Thus, the bit rate for the entire polygon is the sum of all the edge rates that are determined not only by the code table but also by the edge encoding structure. Besides, the approximation quality depends largely on the ability of edge encoding structure to represent the contour characteristics. Therefore, the edge encoding structure plays a vital role in both compression and description performances.

There is an inherent limitation in the eight-direction structure. The edge available for this structure should be restricted to intersect the horizontal axis in an angle which is an integer multiple of  $\pi/4$  (named as restricted edge). The edge that is not in one of these eight restricted directions (named as unrestricted edge) cannot be selected, despite that the edge distortion is no larger than the given admissible distortion. Consequently, it results in the following two problems, as shown in Figs. 1(a) and 1(b):

1. From a data compression perspective, it is a waste of bits, as a large number of short restricted edges are needed when the contour segments are not exactly in eight restricted directions.

\*Address all correspondence to: Zhongyuan Lai, E-mail: laizhy@jhun.edu.cn



**Fig. 1** Comparison results for the Neck region of the 31st frame of the *MissAmerica.qcif* sequence with  $D_{\max} = 1$  pel. Here, we select the upper left corners as the first vertices and use the run length codes to encode the edge. (a) Eight-direction structure with  $VN = 24$ ,  $R = 173$  bits; (b) zoom-in on the portions highlighted by the rectangle in (a); (c) eight-sector structure with  $VN = 12$ ,  $R = 148$  bits; (d) 16-sector structure with  $VN = 13$ ,  $R = 141$  bits. As we can see, eight-direction structure produces a host of redundant short edges, so it is a waste of bits; our two arbitrary direction structures can avoid these redundant short edges, so they save a large number of bits as well as produce more compact results (Notations— $D_{\max}$ : admissible distortion;  $VN$ : vertex number;  $R$ : encoding bits; Legends—dashed line: original contour; solid line: approximated polygon; asterisk: vertex).

- From a pattern recognition perspective, it cannot well describe the object contour, as a large number of selected vertices may not be at or near the corners of the object contour when the contour segments are not exactly in the eight restricted directions.

The above phenomena motivate us to relax the edge direction restriction. To achieve this, we propose two arbitrary direction edge encoding structures, called the 8- and the 16-sector structures, which can encode both the restricted and unrestricted edges. First, we partition the digital coordinates into 8 or 16 sectors, and then decompose the encoding edge into a short component and a long component. After that, we encode the sector information with a fixed length code (FLC), the short and the long components with either the run length codes (RLC) or the variable length codes (VLC) in a different way, providing a further encoding bit reduction. We seamlessly embed these structures into the ORD optimal vertex-based shape coding algorithms with various parameter setups to improve their RD performance as well as to achieve better description results. Some representative results produced by our proposals are shown in Figs. 1(c) and 1(d).

The rest of this paper is organized as follows. Section 2 introduces the ORD optimal shape coding framework, the related enhancements, and the traditional eight-direction edge encoding structure. Section 3 presents the 8- and the 16-sector edge encoding structures, respectively. Full analyses of the experimental results on both shape coding and hand gesture recognition are provided in Sec. 4. Finally, conclusions and future work are given in Sec. 5.

## 2 Related Work

### 2.1 Operational Rate-Distortion Optimal Shape Coding

Let the ordered set  $C = \{c_0, c_1, \dots, c_{N_C-1}\}$  denote the connected contour, where  $c_i$  is the  $i$ 'th pixel of  $C$ ,  $N_C$  is the total number of pixels in  $C$ , and  $c_0 = c_{N_C-1}$  for a closed contour. Let the ordered set  $A = \{a_{0,0}, a_{1,0}, a_{1,1}, \dots, a_{N_C-1,0}\}$  denote the admissible vertex set, where  $a_{i,0} = c_i$ ,  $a_{i,m}$  is the  $m$ 'th admissible vertex associated with  $c_i$ , and  $N(i)$  is the number of admissible vertices in  $A$  associated to  $c_i$ . Let  $L(i)$  denote the length of the sliding window in pixel along the contour, starting at contour point  $c_i$ . Let the two-tuple set  $E_A = \{(a_{i,m}, a_{j,n}) \in A^2: \forall i < j \leq i + L(i)\}$  denote the candidate edges for polygonal approximation. Let the ordered set  $P = \{p_0, p_1, \dots, p_{N_P-1}\}$  denote the polygon used to approximate  $C$ , where  $p_k$  is the  $k$ 'th vertex of  $P$ ,  $N_P$  is the total number of vertices in  $P$ , and  $P \subseteq A$ . Let  $r(p_{k-1}, p_k)$  denote the bit rate of the edge  $(p_{k-1}, p_k)$  and  $R(P)$  the bit rate for the entire polygon. Let  $d(p_{k-1}, p_k)$  denote the distortion of the edge  $(p_{k-1}, p_k)$ ,  $D(P)$  the polygon distortion, and  $D_{\max}$  the admissible distortion. The ORD shape coding finds an optimal polygon  $P$  in the admissible vertex set  $A$  which can be encoded with the lowest bit rate for a given admissible distortion  $D_{\max}$ .<sup>8</sup> It can be formulated as follows:

$$\min_P R(P) \quad \text{s.t.} \quad D(P) \leq D_{\max}, \quad (1)$$

where  $R(P)$  is the sum of all the edge rates and  $D(P)$  is the maximum edge distortion.

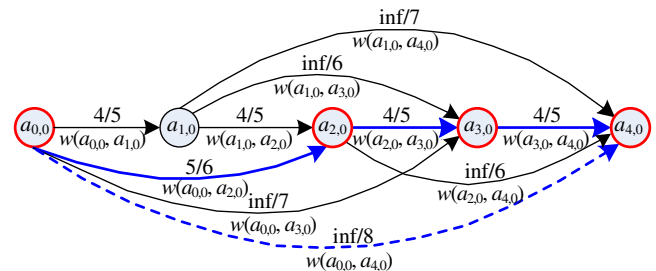
Formulation (1) is virtually a shortest path problem in a weighted DAG  $G = (A, E_A, w)$ , as shown in Fig. 2.<sup>9</sup> A valid path of order  $K$  from vertex  $v_0$  to a vertex  $v_K$  is an ordered set  $\{v_0, v_1, \dots, v_K\}$ , such that  $(v_{k-1}, v_k) \in E_A$  for  $k = 1, 2, \dots, K$ , whose length is defined as follows:

$$\sum_{k=1}^K w(v_{k-1}, v_k), \quad (2)$$

where  $w(u, v)$  is a weight function of the edge  $(u, v)$  defined as follows:

$$w(u, v) = \begin{cases} \infty: & d(u, v) > D_{\max} \\ r(u, v): & d(u, v) \leq D_{\max} \end{cases}. \quad (3)$$

The above definition of the weight function leads to the length of infinity for every path, which includes an edge with



**Fig. 2** Modeled DAG and the shortest path of Fig. 1(d) (Notation—arrow: candidate edge with weights given by the eight-direction/eight-sector structure).

the distortion above  $D_{\max}$ . Therefore, the shortest path will not include this edge. Every finite-length path that starts at vertex  $a_{0,0}$  and ends at vertex  $a_{N_C-1,0}$  results in a path length equal to the rate of the polygon it represents. Therefore, the shortest of all these paths corresponds to the polygon with the smallest bit rate, which is the solution to the problem in (1).

Let  $R^*(a_{i,m})$  represents the minimum rate from  $a_{0,0}$  to  $a_{i,m}$  via a polygon approximation. Let  $q(a_{j,n})$  be the back pointer of  $a_{j,n}$ , which is used to remember the optimal path. Then, the solution of the shortest path can be found efficiently by the DAG shortest path algorithm<sup>21</sup> formalized in Algorithm 1.

We will explain how Algorithm 1 works. In lines 1 and 2, the admissible vertex set and the length of the sliding window length for each contour point are calculated. In line 3, the rate for encoding the starting point of the contour is assigned to the rate of the first polygon vertex, and the rate for reaching any of the admissible vertex is set to infinity. The “for loops” in lines 4 and 5 select the start vertex of a polygon edge and the “for loops” in lines 6 and 7 select its end vertex within the sliding window of the start vertex. The lines 8–10 are used to calculate the edge distortion and the edge rate, by which the edge weight can be determined. The most important part of this algorithm is the comparison in line 11. Here, it tests whether the new minimum rate,  $R^*(a_{i,m}) + w(a_{i,m}, a_{j,n})$ , to reach admissible vertex  $a_{j,n}$ , given that the last vertex was  $a_{i,m}$ , is smaller than the smallest minimum rate used so far to reach  $a_{j,n}$ ,  $R^*(a_{j,n})$ . If this minimum rate is indeed smaller, then it is assigned as the new smallest minimum rate to reach admissible vertex  $a_{j,n}$ ,  $R^*(a_{j,n}) = R^*(a_{i,m}) + w(a_{i,m}, a_{j,n})$ , and the back pointer of  $a_{j,n}$ ,  $q(a_{j,n})$  is assigned to point to  $a_{i,m}$  since this is the previous vertex used to achieve  $R^*(a_{j,n})$ . This algorithm leads to the optimal solution because when the rate  $R^*(a_{i,m})$  of a vertex  $a_{i,m}$  is given, then the selection of the future vertices  $a_{j,n}$ ,  $i < j \leq L(i)$  is independent of the selection of the past vertices  $a_{k,l}$ ,  $0 \leq k < m$ .

## 2.2 Enhancements to the Operational Rate-Distortion Optimal Shape Coding

As we can see, the major problem in the ORD optimal shape coding algorithm is that the optimality is contingent on the chosen admissible vertex set,<sup>1,9,10</sup> sliding window strategy,<sup>1,10,11</sup> distortion measurement,<sup>12–16</sup> code table,<sup>17–19</sup> and the edge encoding structure. The former four parameters have been intensively investigated to relax their constraints and improve the shape coding performance. Here, we provide a brief summary.

1. *Admissible Vertex Set*: The goal of this research is to relax the vertex location restriction. In Ref. 1, the fixed-width admissible vertex set was introduced. Recently, the variable-width admissible vertex set has been proposed,<sup>10</sup> providing another free degree for the RD optimization.<sup>10</sup>
2. *Sliding Window Strategy*: The goal of this research is to relax the edge length restriction. In Ref. 1, the prescribed sliding window length selection strategy was introduced. Recently, the automatic strategies, including the adaptive sliding window strategy<sup>10</sup> and generalized sliding window strategy,<sup>11</sup> have been proposed,

### Algorithm 1 The ORD optimal shape coding algorithm.

#### Inputs:

$C = \{c_0, c_1, \dots, c_{N_C-1}\}$ : the ordered set of contour points;

$D_{\max}$ : the admissible distortion;

#### Output:

$P = \{p_0, p_1, \dots, p_{N_C-1}\}$ : the ordered set of selected vertices.

- 1: Calculate the ordered admissible vertex set  
 $A = \{a_{0,0}, a_{1,0}, a_{1,1}, \dots, a_{N_C-1,0}\}$ ;
- 2: Calculate the length of the sliding window  $L(i)$ ,  $i = 0, 1, \dots, N_C - 2$ ;
- 3: Initialize  $R^*(a_{0,0}) = r(p_{-1}, p_0)$  and  $R^*(a_{i,m}) = \infty$  for  $i = 1, 2, \dots, N_C - 1$  and  $m = 0, 1, \dots, N(i)$ ;
- 4: **for**  $i = 0, 1, \dots, N_C - 2$  **do**
- 5:   **for**  $m = 0, 1, \dots, N(i)$  **do**
- 6:     **for**  $j = i + 1, i + 2, \dots, i + L(i)$  **do**
- 7:       **for**  $n = 0, 1, \dots, N(j)$  **do**
- 8:         Calculate the edge distortion  $d(a_{i,m}, a_{j,n})$ ;
- 9:         Calculate the edge rate  $r(a_{i,m}, a_{j,n})$ ;
- 10:         Assign the edge weight  $w(a_{i,m}, a_{j,n})$  based on definition (3);
- 11:         **if**  $R^*(a_{i,m}) + w(a_{i,m}, a_{j,n}) < R^*(a_{j,n})$  **then**
- 12:              $R^*(a_{j,n}) = R^*(a_{i,m}) + w(a_{i,m}, a_{j,n})$ ;
- 13:              $q(a_{j,n}) = a_{i,m}$ ;
- 14:         **end if**
- 15:       **end for**
- 16:     **end for**
- 17:   **end for**
- 18: **end for**
- 19: Trace back from  $q(a_{N_C-1,0})$  until meet  $a_{0,0}$  and reverse the path to obtain  $P$ ;
- 20: **return**  $P$ .

providing better tradeoffs among the RD, computational, and memory efficiency.

3. *Distortion Measurements*: As the most time-consuming operation, the goal of this research is to improve its visual consistency while keeping the computing cost as low as possible. In Ref. 1, the shortest absolute distance (SAD) and the fixed-width distortion band (DB) were introduced. In Ref. 12, the variable-width DB was presented. In Ref. 13, the accurate distortion



measurement for shape coding (ADMSC) was presented to improve the accuracy of SAD and DB, and in Ref. 14, its fast algorithm was presented using the chord-length parameterization. Recently, the perceptual distortion measure has been proposed to improve the visual consistency of SAD and DB from the perspective of vision psychology.<sup>15</sup> See Ref. 16, for a contemporary review.

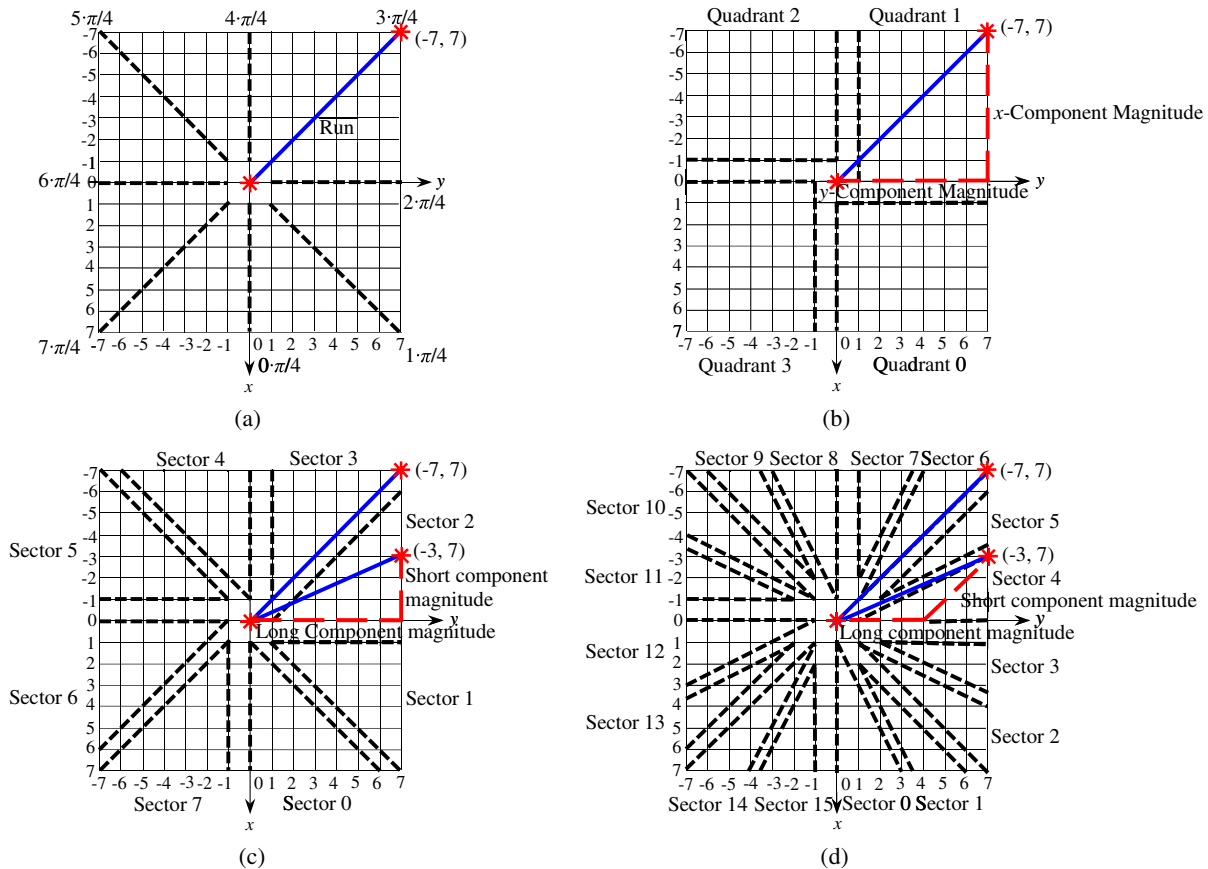
4. *VLC Table Optimization:* The goal of this research is to remove the conditioning of the ORD optimal solution on an *ad hoc* VLC table. In Ref. 17, the VLC optimization for the intra shape coding was proposed using the unconditional symbol probabilities of encoding edges. Then, it was extended to the inter shape coding<sup>18</sup> and the scalable shape coding<sup>19</sup> using the conditional symbol probabilities given the prior knowledge from the previous encoded frames and layers, respectively.

### 2.3 Traditional Eight-Direction Edge Encoding Structure

While intensive investigations have been made on the above four parameters, little attention has been paid to the edge encoding structure. Almost all the vertex-based ORD

optimal shape coding algorithms employ the eight-direction edge encoding structure.<sup>20</sup> In this structure, the eight-connected chain code and the run-length encoding are combined by representing the edge between two admissible vertices by an angle  $\alpha_{8\text{-dir}}$  and a run  $\beta_{8\text{-dir}}$ , which form the symbol  $(\alpha_{8\text{-dir}}, \beta_{8\text{-dir}})$ , as illustrated in Fig. 3(a). Each of the possible symbols  $(\alpha_{8\text{-dir}}, \beta_{8\text{-dir}})$  gets a probability assigned and the resulting stream of the polygon  $P$ 's edges can be optimally encoded using these probabilities.

In practical implementations, the design of code is under the hypothesis that the probability mass function of  $(\alpha_{8\text{-dir}}, \beta_{8\text{-dir}})$  is separable<sup>20</sup> and  $\alpha_{8\text{-dir}}$  is uniformly distributed over all the eight restricted directions. Thus,  $\alpha_{8\text{-dir}}$  can be encoded separately and the optimal code for  $\alpha_{8\text{-dir}}$  is a 3-bit FLC. There are two main codes for  $\beta_{8\text{-dir}}$ . One is based on the hypothesis that  $\beta_{8\text{-dir}}$  is geometrically distributed  $[P(\beta_{8\text{-dir}} = j) = (1 - p)/p^{j-1}, j \geq 1]$  with a parameter  $p = 0.5$ . In this case, the optimal code for  $\beta_{8\text{-dir}}$  is a RLC with  $\beta_{8\text{-dir}} - 1$  zeros and a final "1." Note that any positive integer value of  $\beta_{8\text{-dir}}$  is codable; therefore, the code has no edge length restriction. The other is based on the hypothesis that  $\beta_{8\text{-dir}}$  is piecewise uniformly distributed. In this case, the optimal code for  $\beta_{8\text{-dir}}$  is a VLC, which has the general form  $[YX]$ , where  $Y$  is the length of  $X$ . Note that the largest possible value of  $\beta_{8\text{-dir}}$  has to be a finite positive integer  $\epsilon$ .



**Fig. 3** Illustration of (a) the angle and the run of the eight-direction structure, (b) the quadrant number, the x-component magnitude, and the y-component magnitude of the quadrant structure, (c) the sector number, the short component magnitude, and the long component magnitude of the eight-sector structure, and (d) the sector number, the short component magnitude, and the long component magnitude of the 16-sector structure (Legend—solid line: encoding edge; asterisk: vertex; long dashed line: edge component; dashed line in (a): admissible positions for the next vertex given the current vertex at the origin; dashed line in (b): quadrant boundary; dashed line in (c) and (d): sector boundary).

A smaller  $\varepsilon$  often results in fewer bits for each edge, but requires more edges for the entire polygon. Conversely, a larger  $\varepsilon$  often results in fewer edges, but requires more bits for each edge. Usually,  $\varepsilon = 15$  makes a good balance between the edge rate and the edge number over a wide range of contour characteristics and admissible distortions.<sup>11</sup> Thus, 15 codewords are needed. Reference 20 gives a typical example of such code table that  $Y$  and  $X$  are a 2-bit FLC and a  $\lfloor \log_2 \beta_{8\text{-dir}} \rfloor$ -bit FLC, respectively, where  $\lfloor \cdot \rfloor$  is the floor operator.

To illustrate the eight-direction structure-based edge encoding scheme for encoding an edge, consider a restricted edge  $(-7,7)$  (for simplicity, we use the coordinates of the ending point as the coordinates of the edge when this edge starts at the origin, similarly hereinafter), as shown in Fig. 3(a). Its angle and run take the values of  $3 \cdot \pi/4$  and 7, which are encoded with a 3-bit FLC and a 7-bit RLC (a 4-bit VLC), resulting in 10 bits (7 bits) in total.

The above eight-direction structure-based edge encoding scheme is quite simple and easy to implement. However, the inherent limitation can be clarified as follows.

From a structural perspective, the uniquely decodable code for each edge consists of two parts representing  $\alpha_{8\text{-dir}}$  and  $\beta_{8\text{-dir}}$ , and the total length of this code is  $3 + \beta_{8\text{-dir}}$  when RLC is used or  $5 + \lfloor \log_2 \beta_{8\text{-dir}} \rfloor$  when VLC is used. For simplicity, assume that an edge of run  $\beta_{8\text{-dir}}$  approximates a contour segment of  $\beta_{8\text{-dir}}$  pixels. Then, the efficiency of this code, usually measured by the bits per contour pixel, is  $3/\beta_{8\text{-dir}} + 1$  ( $\beta_{8\text{-dir}} = 1, 2, \dots$ ) or  $(5 + \lfloor \log_2 \beta_{8\text{-dir}} \rfloor)/\beta_{8\text{-dir}}$  ( $\beta_{8\text{-dir}} = 1, 2, \dots, 15$ ). Note that both sequences are increasing as  $\beta_{8\text{-dir}}$  decreases; therefore, the shorter the run, the less efficient these codes are. Figures 1(a) and 1(b) show that many short restricted edges are needed for the contour segments which are not exactly in eight restricted directions. Therefore, the eight-direction edge encoding structure is inefficient.

From a description perspective, a good shape description should make the proportion of selected vertices at or near the corners of the original shape contour as high as possible so as to benefit the object-oriented applications.<sup>22</sup> However, when the contour segments are not exactly in eight restricted directions, the eight-direction structure not only consumes a large number of closely spaced vertices, but also strengthens the contour quantization error and noise disturbances. This affects the subjective experiences of reconstruction contours as well as degrades the performance of the object-oriented applications.

Further, to reveal the problem of the existing eight-direction structure quantitatively from a systematic perspective, we map the contour segment in Fig. 1(b) into a weighted

$$r(\alpha_{8\text{-dir}}, \beta_{8\text{-dir}}) = \begin{cases} 3 + \beta_{8\text{-dir}} = 3 + \max\{\beta_{\text{quad}}, \gamma_{\text{quad}}\}: \alpha_{8\text{-dir}} = k \cdot \pi/4, & k = 0, 1, \dots, 7 \\ \infty: \text{otherwise} \end{cases} \quad (5)$$

Compared Eq. (4) with Eq. (5), we have  $r(\alpha_{\text{quad}}, \beta_{\text{quad}}, \gamma_{\text{quad}}) = 3 + 2\beta_{8\text{-dir}}$  when  $\beta_{\text{quad}} = \gamma_{\text{quad}} = \beta_{8\text{-dir}}$ . Thus, although the quadrant structure can change the unrestricted edge rate from infinity to a finite value, it increases the diagonal direction edge rate from  $3 + \beta_{8\text{-dir}}$  to  $3 + 2\beta_{8\text{-dir}}$ , resulting in  $\beta_{8\text{-dir}}$  bits increment. Assume that an edge of run  $\beta_{8\text{-dir}}$  approximates a contour segment

DAG, as shown in Fig. 2. It is observed that all 10 edge distortions uphold the admissible distortion, but 5 out of 10 edge weights are assigned infinity due to the direction restriction, making the shortest path in DAG quite long and thus the bit rate for the entire polygon much larger. Therefore, we should relax the direction restriction to increase the number of available edges. The following section follows this idea.

### 3 Arbitrary Direction Edge Encoding Structures

#### 3.1 Eight-Sector Edge Encoding Structure

A simple and intuitive structure for an edge of arbitrary direction consists of a quadrant number  $\alpha_{\text{quad}}$ , an  $x$ -component magnitude  $\beta_{\text{quad}}$ , and an  $y$ -component magnitude  $\gamma_{\text{quad}}$ , which form the symbol  $(\alpha_{\text{quad}}, \beta_{\text{quad}}, \gamma_{\text{quad}})$ . The quadrant number  $\alpha_{\text{quad}}$  is defined as: quadrant 0 is the set of  $(x, y)$  such that  $x \geq 1, y \geq 0$ , quadrant 1:  $x \leq 0, y \geq 1, \dots$ , and quadrant 3:  $x \geq 0, y \leq -1$ , as illustrated in Fig. 3(b). Assume that  $\alpha_{\text{quad}}, \beta_{\text{quad}}$ , and  $\gamma_{\text{quad}}$  are independently distributed;  $\alpha_{\text{quad}}$  is uniformly distributed over all the four quadrants;  $\beta_{\text{quad}}$  and  $\gamma_{\text{quad}}$  are geometrically or piecewise uniformly distributed. Then,  $\alpha_{\text{quad}}$  can be separately and optimally encoded with a 2-bit FLC and  $\beta_{\text{quad}}$  and  $\gamma_{\text{quad}}$  can be optimally encoded with RLCs or VLCs. The procedures to implement the quadrant structure-based edge encoding scheme are as follows:

1. determine the quadrant number  $\alpha_{\text{quad}}$  and encode it with a 2-bit FLC;
2. if  $\alpha_{\text{quad}}$  is even, increase  $\beta_{\text{quad}}$  by 1 to make the value of  $\beta_{\text{quad}}$  range from 1; If  $\alpha_{\text{quad}}$  is odd, increase  $\gamma_{\text{quad}}$  by 1 to make the value of  $\gamma_{\text{quad}}$  range from 1;
3. if RLC is selected, encode  $\beta_{\text{quad}}$  with  $\beta_{\text{quad}} - 1$  zeros and a final "1" and encode  $\gamma_{\text{quad}}$  with  $\gamma_{\text{quad}} - 1$  zeros and a final "1," respectively. If VLC is selected, similarly with the run encoding in the eight-direction structure-based scheme, encode  $\beta_{\text{quad}}$  with a  $(2 + \lfloor \log_2 \beta_{\text{quad}} \rfloor)$ -bit VLC and encode  $\gamma_{\text{quad}}$  with a  $(2 + \lfloor \log_2 \gamma_{\text{quad}} \rfloor)$ -bit VLC.

Now we compare the performance of this quadrant structure with the traditional eight-direction one. If RLC is selected, for the quadrant structure, the bit rate for the symbol  $(\alpha_{\text{quad}}, \beta_{\text{quad}}, \gamma_{\text{quad}})$ , denoted by  $r(\alpha_{\text{quad}}, \beta_{\text{quad}}, \gamma_{\text{quad}})$ , is

$$r(\alpha_{\text{quad}}, \beta_{\text{quad}}, \gamma_{\text{quad}}) = 3 + \beta_{\text{quad}} + \gamma_{\text{quad}}, \quad (4)$$

while for the eight-direction structure, we have  $\beta_{8\text{-dir}} = \max\{\beta_{\text{quad}}, \gamma_{\text{quad}}\}$  for the restricted edge, thus the bit rate for the symbol  $(\alpha_{8\text{-dir}}, \beta_{8\text{-dir}})$ , denoted by  $r(\alpha_{8\text{-dir}}, \beta_{8\text{-dir}})$ , is

of  $\beta_{8\text{-dir}}$  pixels. Then, for the the diagonal direction edge, the quadrant structure changes the code efficiency from  $3/\beta_{8\text{-dir}} + 1$  to  $3/\beta_{8\text{-dir}} + 2$ , resulting in 1 bit increment per contour pixel. Such a large cost makes the quadrant structure unfeasible in practice. With a similar comparison method, we can draw the same conclusion for the VLC case.

To illustrate this deficiency in the quadrant structure-based edge encoding scheme, reconsider the restricted edge  $(-7,7)$ , as shown in Fig. 3(b). Its quadrant numbers,  $x$ -component magnitude and  $y$ -component magnitude, take the values of 2, 7, and 7, which are encoded with a 2-bit FLC, an 8-bit RLC (a 5-bit VLC), and a 7-bit RLC (a 4-bit VLC), resulting in 17 bits (11 bits) in total. This is 7 bits (4 bits) more than those used by eight-direction structure.

The reason that the quadrant structure does not work well is that there exists dependency between the  $x$ - and the  $y$ -component magnitudes and the independency assumption is unreasonable. It motivates us to make a good use of this dependency. To achieve this aim, we design an edge encoding structure consisting of a sector number  $\alpha_{8\text{-sec}}$ , a short component magnitude  $\beta_{8\text{-sec}}$ , and a long component magnitude  $\gamma_{8\text{-sec}}$ , which form the symbol  $(\alpha_{8\text{-sec}}, \beta_{8\text{-sec}}, \gamma_{8\text{-sec}})$ . From Ref. 2, we define the sector number  $\alpha_{8\text{-sec}}$  as: sector 0 is the set of  $(x, y)$  such that  $0 \leq y < x$ , sector 1:  $1 \leq x \leq y$ , and sector 7:  $1 \leq -y \leq x$ , as illustrated in Fig. 3(c). The short component is the smaller component of the edge's  $x$ - and  $y$ -components, while the long component is the larger component of them. Because the long component magnitude  $\gamma_{8\text{-sec}}$  is not smaller than the short component magnitude  $\beta_{8\text{-sec}}$ , we can encode  $\gamma_{8\text{-sec}}$  with differential encoding method. In this way, we can make a good use of the dependency between the  $x$ - and the  $y$ -component magnitudes, and allow a further reduction in the number of bits used.

Let  $\delta_{8\text{-sec}}$  denote the difference between  $\beta_{8\text{-sec}}$  and  $\gamma_{8\text{-sec}}$ , i.e.,  $\delta_{8\text{-sec}} = \gamma_{8\text{-sec}} - \beta_{8\text{-sec}}$ . In our practical implementation, we design the code under the hypothesis that the probability mass function of  $(\alpha_{8\text{-sec}}, \beta_{8\text{-sec}}, \gamma_{8\text{-sec}})$  is separable and  $\alpha_{8\text{-sec}}$  is uniformly distributed over all the eight sectors. Thus,  $\alpha_{8\text{-sec}}$  can be separately and optimally encoded with a 3-bit FLC. Similarly with the assumption about the run in the eight-direction structure-based scheme,<sup>20</sup> we simply assume that both  $\beta_{8\text{-sec}}$  and  $\delta_{8\text{-sec}}$  are geometrically or piecewise uniformly distributed. Therefore, the optimal codes for both  $\beta_{8\text{-sec}}$  and  $\delta_{8\text{-sec}}$  are RLCs or VLCs. The procedures to implement the eight-sector structure-based edge encoding scheme are as follows:

1. determine the sector number  $\alpha_{8\text{-sec}}$  and encode it with a 3-bit FLC;
2. determine the short component magnitude  $\beta_{8\text{-sec}}$  and the long component magnitude  $\gamma_{8\text{-sec}}$  according to  $\alpha_{8\text{-sec}}$ , and calculate the difference  $\delta_{8\text{-sec}}$ ;
3. if  $\alpha_{8\text{-sec}}$  is even, increase  $\beta_{8\text{-sec}}$  by 1 to make the value of  $\beta_{8\text{-sec}}$  range from 1; if  $\alpha_{8\text{-sec}}$  is odd, increase  $\delta_{8\text{-sec}}$  by 1 to make the value of  $\delta_{8\text{-sec}}$  range from 1;
4. if RLC is selected, encode  $\alpha_{8\text{-sec}}$  with  $\beta_{8\text{-sec}} - 1$  zeros and a final "1" and encode  $\delta_{8\text{-sec}}$  with  $\delta_{8\text{-sec}} - 1$  zeros and a final "1," respectively.

If VLC is selected, similarly with the run encoding in the eight-direction structure-based scheme, the magnitudes of both  $x$ - and  $y$ - components are restricted to a predefined number  $\varepsilon$ . Thus, we should design a series of code tables of the ranges from 1 to 1, 1 to 2, 1 to  $\varepsilon$  for both  $\beta_{8\text{-sec}}$  and  $\delta_{8\text{-sec}}$ . A series of reference code tables with  $\varepsilon = 15$  are shown in Table 1. Then, encode  $\beta_{8\text{-sec}}$  according to

the table of the range from 1 to  $\varepsilon$  and encode  $\delta_{8\text{-sec}}$  according to the table of the range from 1 to  $\varepsilon - \beta_{8\text{-sec}} + 1$ .

Here, we reason why this structure works better than the traditional eight-direction one. If RLC is selected, for the eight-sector structure, the bit rate for the symbol  $(\alpha_{8\text{-sec}}, \beta_{8\text{-sec}}, \gamma_{8\text{-sec}})$ , denoted by  $r(\alpha_{8\text{-sec}}, \beta_{8\text{-sec}}, \gamma_{8\text{-sec}})$ , is

$$r(\alpha_{8\text{-sec}}, \beta_{8\text{-sec}}, \gamma_{8\text{-sec}}) = 4 + \beta_{8\text{-sec}} + \delta_{8\text{-sec}} = 4 + \gamma_{8\text{-sec}}, \quad (6)$$

where  $\gamma_{8\text{-sec}} = \beta_{8\text{-dir}}$  when the encoding edge is in the restricted direction. Comparing Eq. (6) with Eq. (5), we can see that although the eight-sector structure increases the restricted edge rate by 1 bit, it can change the unrestricted edge rate from infinity to an acceptable finite value. Therefore, a large number of bits can be saved in average. With similar comparison method, we can draw the same conclusion for the VLC case.

### 3.2 Sixteen-Sector Edge Encoding Structure

The above eight-sector structure defines the short component magnitude  $\beta_{8\text{-sec}}$  and the long component magnitude  $\gamma_{8\text{-sec}}$  as the smaller and the larger magnitudes of the edge's  $x$ - and  $y$ -components, and encode  $\beta_{8\text{-sec}}$  and the difference  $\delta_{8\text{-sec}}$  in a separate way. If RLC is selected, this structure requires  $4 + \beta_{8\text{-sec}} + \delta_{8\text{-sec}}$  bits. Although it can basically keep the bit rate for restricted edges, it will lead to a relatively larger number of bits for unrestricted edges, especially when both  $\beta_{8\text{-sec}}$  and  $\delta_{8\text{-sec}}$  take large values. The reason for this inefficiency is that there exists dependency between  $\beta_{8\text{-sec}}$  and  $\delta_{8\text{-sec}}$  and the independency assumption is unreasonable. It motivates us to make a good use of this dependency.

To achieve this aim, we design an edge encoding structure consisting of a sector number  $\alpha_{16\text{-sec}}$ , a short component magnitude  $\beta_{16\text{-sec}}$ , and a long component magnitude  $\gamma_{16\text{-sec}}$ , which form a symbol  $(\alpha_{16\text{-sec}}, \beta_{16\text{-sec}}, \gamma_{16\text{-sec}})$ . The sector number  $\alpha_{16\text{-sec}}$  is defined as: sector 0 is the set of  $(x, y)$  such that  $0 \leq 2y < x$ , sector 1:  $1 \leq y < x \leq 2y, \dots$ , and sector 15:  $1 \leq -2y \leq x$ , as illustrated in Fig. 3(d). The short component is the smaller component of the edge's two octant direction components with  $\pi/4$  difference in angle, while the long component is the larger component of them. Because the long component magnitude  $\gamma_{16\text{-sec}}$  is not smaller than the short component magnitude  $\beta_{16\text{-sec}}$ , we can encode  $\gamma_{16\text{-sec}}$  with differential encoding method. Compared Fig. 3(d) with Fig. 3(c), we have  $\beta_{16\text{-sec}} = \min\{\beta_{8\text{-sec}}, \delta_{8\text{-sec}}\}$  and  $\gamma_{16\text{-sec}} = \max\{\beta_{8\text{-sec}}, \delta_{8\text{-sec}}\}$ . Therefore, in this way, we can make a good use of the dependency between the short and the long component magnitudes of the eight-sector structure, and allow a further reduction in the number of bits used.

In our practical implementation, we use the same hypotheses and the corresponding optimal codes as we used in the eight-sector structure. The procedures to implement the 16-sector structure-based edge encoding scheme are as follows:

1. determine the sector number  $\alpha_{16\text{-sec}}$  and encode it with a 4-bit FLC;
2. determine the short component magnitude  $\alpha_{16\text{-sec}}$  and the long component magnitude  $\gamma_{16\text{-sec}}$  according to  $\alpha_{16\text{-sec}}$ , and calculate the difference  $\delta_{16\text{-sec}}$ ;

**Table 1** Variable length encoding table for  $\varepsilon = 15$ . The number of bits is determined by both the range and the value of  $\beta_{8\text{-sec}}$  ( $\beta_{16\text{-sec}}$ ) or  $\delta_{8\text{-sec}}$  ( $\delta_{16\text{-sec}}$ ).

Range	$\beta_{8\text{-sec}}$ ( $\beta_{16\text{-sec}}$ ) or $\delta_{8\text{-sec}}$ ( $\delta_{16\text{-sec}}$ ) value														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
From 1 to 15	2	3	3	4	4	4	4	5	5	5	5	5	5	5	5
From 1 to 14	2	3	3	4	4	4	4	4	5	5	5	5	5	5	
From 1 to 13	2	3	3	4	4	4	4	4	4	5	5	5	5		
From 1 to 12	2	3	3	4	4	4	4	4	4	4	5	5			
From 1 to 11	2	3	3	4	4	4	4	4	4	4	4				
From 1 to 10	2	3	3	3	4	4	4	4	4	4					
From 1 to 9	2	3	3	3	3	4	4	4	4						
From 1 to 8	2	3	3	3	3	3	4	4							
From 1 to 7	2	3	3	3	3	3	3								
From 1 to 6	2	2	3	3	3	3									
From 1 to 5	2	2	2	3	3										
From 1 to 4	2	2	2	2											
From 1 to 3	1	2	2												
From 1 to 2	1	1													
From 1 to 1	0														

3. if  $\alpha_{16\text{-sec}}$  is even, increase  $\beta_{16\text{-sec}}$  by 1 to make the value of  $\beta_{16\text{-sec}}$  range from 1; if  $\alpha_{16\text{-sec}}$  is odd, increase  $\delta_{16\text{-sec}}$  by 1 to make the value of  $\delta_{16\text{-sec}}$  range from 1;
4. if RLC is selected, encode  $\beta_{16\text{-sec}}$  with  $\beta_{16\text{-sec}} - 1$  zeros and a final "1" and encode  $\delta_{16\text{-sec}}$  with  $\delta_{16\text{-sec}} - 1$  zeros and a final "1."

If VLC is selected, the magnitudes of both  $x$ - and  $y$ -components are restricted to  $\varepsilon$ . Thus, we should design a series of code tables of the ranges from 1 to 1, 1 to 2, ..., 1 to  $\varepsilon$  for both  $\beta_{16\text{-sec}}$  and  $\delta_{16\text{-sec}}$ . Table 1 can also be used here:

1. If  $\alpha_{16\text{-sec}}$  is even, encode  $\beta_{16\text{-sec}}$  according to the table of the range from 1 to  $\lfloor(\varepsilon + 1)/2\rfloor$  and encode  $\delta_{16\text{-sec}}$  according to the table of the range from 1 to  $\varepsilon - 2\beta_{16\text{-sec}} + 2$ .
2. If  $\alpha_{16\text{-sec}}$  is odd, encode  $\beta_{16\text{-sec}}$  according to the table of the range from 1 to  $\lfloor\varepsilon/2\rfloor$  and encode  $\delta_{16\text{-sec}}$  according to the table of the range from 1 to  $\varepsilon - 2\beta_{16\text{-sec}} + 1$ .

Here, we reason why this structure with RLC can outperform the eight-sector one from a data compression perspective. For the 16-sector structure, the bit rate for the symbol  $(\alpha_{16\text{-sec}}, \beta_{16\text{-sec}}, \gamma_{16\text{-sec}})$ , denoted by  $r(\alpha_{16\text{-sec}}, \beta_{16\text{-sec}}, \gamma_{16\text{-sec}})$ , is

$$\begin{aligned} r(\alpha_{16\text{-sec}}, \beta_{16\text{-sec}}, \gamma_{16\text{-sec}}) &= 5 + \beta_{16\text{-sec}} + \delta_{16\text{-sec}} \\ &= 5 + \gamma_{16\text{-sec}}, \end{aligned} \quad (7)$$

where  $\beta_{16\text{-sec}} = \min\{\beta_{8\text{-sec}}, \delta_{8\text{-sec}}\}$  and  $\gamma_{16\text{-sec}} = \max\{\beta_{8\text{-sec}}, \delta_{8\text{-sec}}\}$ . Compared Eq. (7) with Eq. (6), we can see that although the 16-sector structure increases the restricted edge rate by 1 bit, it can change the unrestricted edge rate from  $4 + \beta_{16\text{-sec}} + \gamma_{16\text{-sec}}$  to  $5 + \gamma_{16\text{-sec}}$ , resulting in the  $\beta_{16\text{-sec}} - 1$  bits decrement. Assume that the edge direction is uniformly distributed, then from Fig. 3(d), we can see that the probability that  $\beta_{16\text{-sec}} \geq 2$  will occur is far more than the probability that  $\beta_{16\text{-sec}} = 0$  will occur. Therefore, a large number of bits can be saved in average.

To illustrate this advantage, consider an unrestricted edge  $(-3, 7)$ , as shown in Figs. 3(c) and 3(d). For the eight-sector structure, its sector number, short component magnitude, and long component magnitude take the values of 2, 3, and 7, which are encoded with a 3-bit FLC, a 4-bit RLC (4-bit VLC), and a 4-bit RLC (4-bit VLC), resulting in 11 bits in total. For the 16-sector structure, they take the values of 4, 3, and 4, which are encoded with a 4-bit FLC, a 4-bit RLC (a 3-bit VLC), and a 1-bit RLC (a 2-bit VLC), resulting in 9 bits in total. Thus, two bits are saved.

However, this bit reduction will degrade the ability in contour description. To look deep into this phenomenon, we plot



the isorate contours for both the eight- and the 16-sector structures on Fig. 4, i.e., the traces of the ending points of the encoding edges having the same starting point and rate. For simplicity, assume that an edge of long component magnitude  $\gamma_{8\text{-sec}}$  approximates a contour segment of  $\gamma_{8\text{-sec}}$  pixels. We can see that for the eight-sector structure, the edges of the same rate have the same  $\gamma_{8\text{-sec}}$ , thus can approximate contour segments of the same length in pixel. Therefore, for the eight-sector structure, the codes for the edges of the same rate have the same efficiency in terms of the bits per contour pixel. Instead, for the 16-sector structure, the edges of the same rate have different  $\gamma_{8\text{-sec}}$ . To be more specific, the edges of directions  $k \cdot \pi/2 \pm \arctan(1/2)$  ( $k = 0, 1, 2, 3$ ) have the maximum length over all the directions (named as preferential direction), while the edges of the restricted directions have the minimum length (named as nonpreferential direction), as shown in Fig. 4(b). Thus, the edges of the same rate but of different directions will approximate contour segments of the different lengths in pixel. Therefore, for the 16-sector structure, the codes for the edges of the same rate have different efficiencies, i.e., the codes for edges in or near the preferential directions have relatively higher efficiencies, whereas the codes for the edges in or near the nonpreferential directions have relatively lower efficiencies. This may bias the edge selection toward preferential directions via the ORD optimal framework, thereby giving false corners when contour segments are in or near the nonpreferential directions. We call it direction bias effect.

To illustrate this effect, we take the approximation of the horizontal contour segment from point  $a_{i,m}(0,0)$  to point  $a_{j,n}(0,12)$  shown in Fig. 4(b) as an example. For the eight-sector structure, among all approximations, the one directly using edge  $(a_{i,m}, a_{j,n})$  is of the most efficiency. However, for the 16-sector structure, according to Eq. (7), nonpreferential direction edge  $(a_{i,m}, a_{j,n})$  will consume 17 bits, whereas two preferential direction edges  $(a_{i,m}, a_{k,l})$  and  $(a_{k,l}, a_{j,n})$  consume 8 bits and 8 bits, respectively, resulting in 16 bits in total. Thus, the ORD optimal framework may select these two edges rather than edge  $(a_{i,m}, a_{j,n})$  to

approximate the horizontal contour segment, which gives a false corner  $a_{k,l}(3,6)$ . This effect may significantly affect the performance of related object-oriented applications, which will be further displayed in the next section.

## 4 Experimental Results

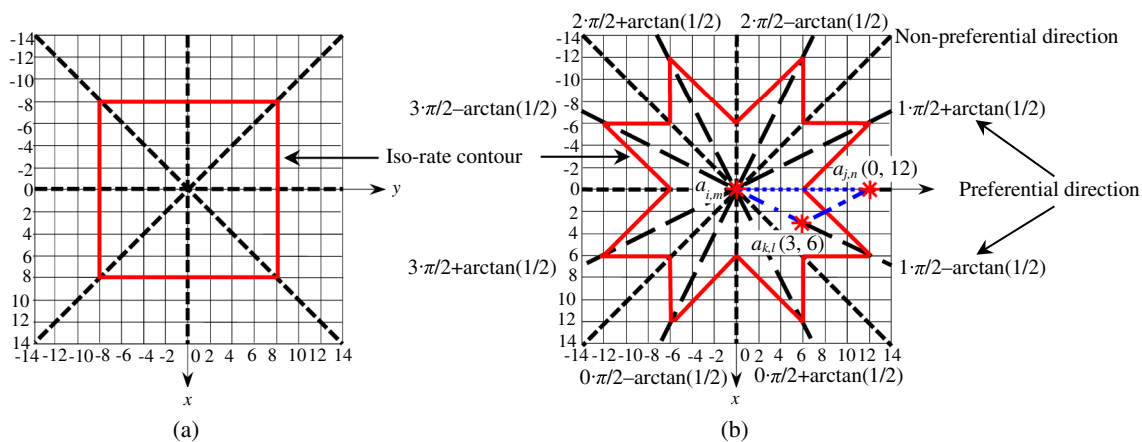
To analyze the performance of the ORD optimal framework with our two arbitrary direction edge encoding structures in both data compression and pattern recognition fields, various ORD optimal shape coding algorithms with different parameter configurations are applied to both shape coding and hand gesture recognition. The ADMSC is chosen for contour distortion measurement.<sup>16</sup> To clarify the nomenclature adopted, the following three-parameter notation is used: Admissible vertex band type, Edge encoding structure type, and Code table type.

Admissible vertex band (AVB) type refers to whether the AVB of width 1 pel is used; Edge encoding structure type refers to the choice of 8-direction, 8-sector, and 16-sector structure; and Code table type refers to the choice of RLC and VLC (the default type is RLC). Therefore, for instance, Basic-8-Direction-RLC means that the algorithm is based on the eight-direction structure with RLC where the AVB technique is not used.

### 4.1 Arbitrary Direction Edge Encoding Structures for Shape Coding

For the RD performance assessments, five MPEG-4 binary shape sequences, namely Weather.qcif, News.qcif, Stefan.sif, Children.sif, and Forman.cif, are used. These sequences have various spatial and temporal resolutions.

Figure 5(a) shows the cumulative RD curves generated with different ORD optimal algorithms. As expected, our two structures always outperform the traditional eight-direction structure, with up to 48.9% bit savings. Figure 5(b) compares the number of available edges between eight-direction and our arbitrary direction structures on Stefan sequence when  $D_{\max} = 1.5$  pels. The numbers of edges available for our arbitrary direction structures without AVB and with AVB are 3–6.3 and 3.5–5.5 times of those available for



**Fig. 4** (a) Isorate contour of the eight-sector structure when  $R = 12$  bits, (b) isorate contour of the 16-sector structure when  $R = 11$  bits, where the direction bias effect happens when contour segments are in or near the nonpreferential directions (Legend—solid line: isorate contour; dashed line in (a): restricted direction; long dashed line in (b): preferential direction; dashed line in (b): nonpreferential direction; dotted line in (b): encoding edge for the 8-sector structure; dashed dotted line in (b): encoding edge for the 16-sector structure; asterisk in (b): vertex).

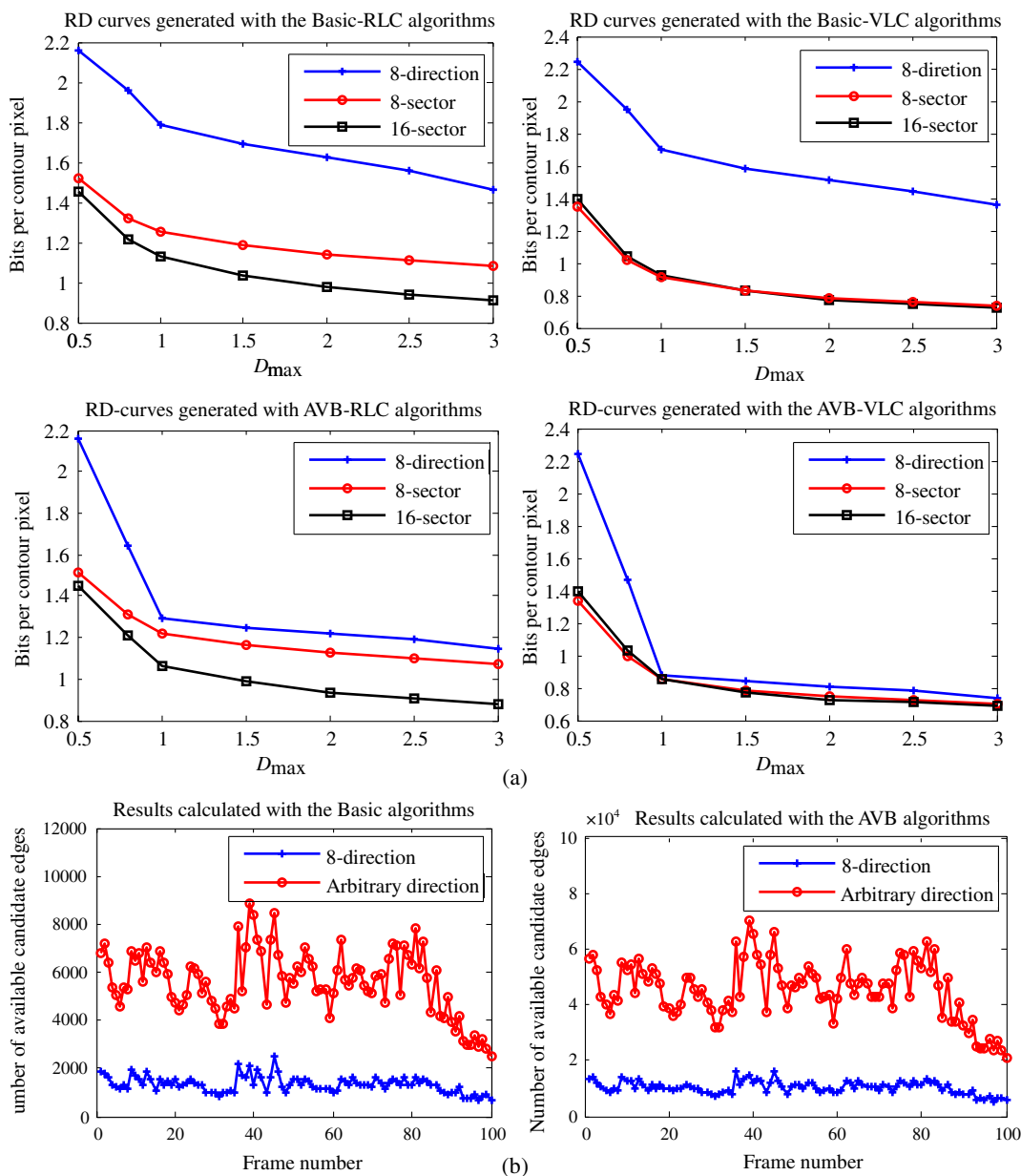
the eight-direction structure. These additional available edges provide the ORD optimal framework more opportunities to find a shorter shortest paths in a weighted DAG with much less infinity weights. That is why our structures can result in much fewer encoding bits than the eight-direction one, which is consistent with the analysis given in Sec. 2.3. It also indicates how important the direction relaxation is from a compression perspective.

Figure 5(a) also reveals that the coding gain varies with different parameter configurations. Here, we only point out the variations with two different configurations and give explanations as follows.

1. The coding gains of our two structures are almost the same when VLC is applied. This is because the bits saved by the 16-sector structure compared with the

8-sector structure come from the range and the value reductions of the short and the difference component. According to Table 1, these reductions may only lead to a few bit reduction in average to balance the additional one bit increment from the sector number, thus no further bit will be saved in average.

2. The coding gains of our two structures are much better without AVB when  $D_{\max} \geq 1$  pel. This is because for the AVB configuration, vertices can be selected outside the original contour, which makes a large number of contour segments with the direction approximately along the restricted directions available to be encoded by one restricted edges. Thus, the eight-direction structure becomes much more efficient. As a result, the coding gains of our two structures decreases.



**Fig. 5** Compression performance comparisons. (a) Cumulative RD results from five sequences calculated by various ORD optimal algorithms; (b) number of available edges calculated frame by frame on the Stefan sequence when  $D_{\max} = 1.5$  pels.

For the description assessments, the Stefan object of the 100th frame of the Stefan sequence has been used.

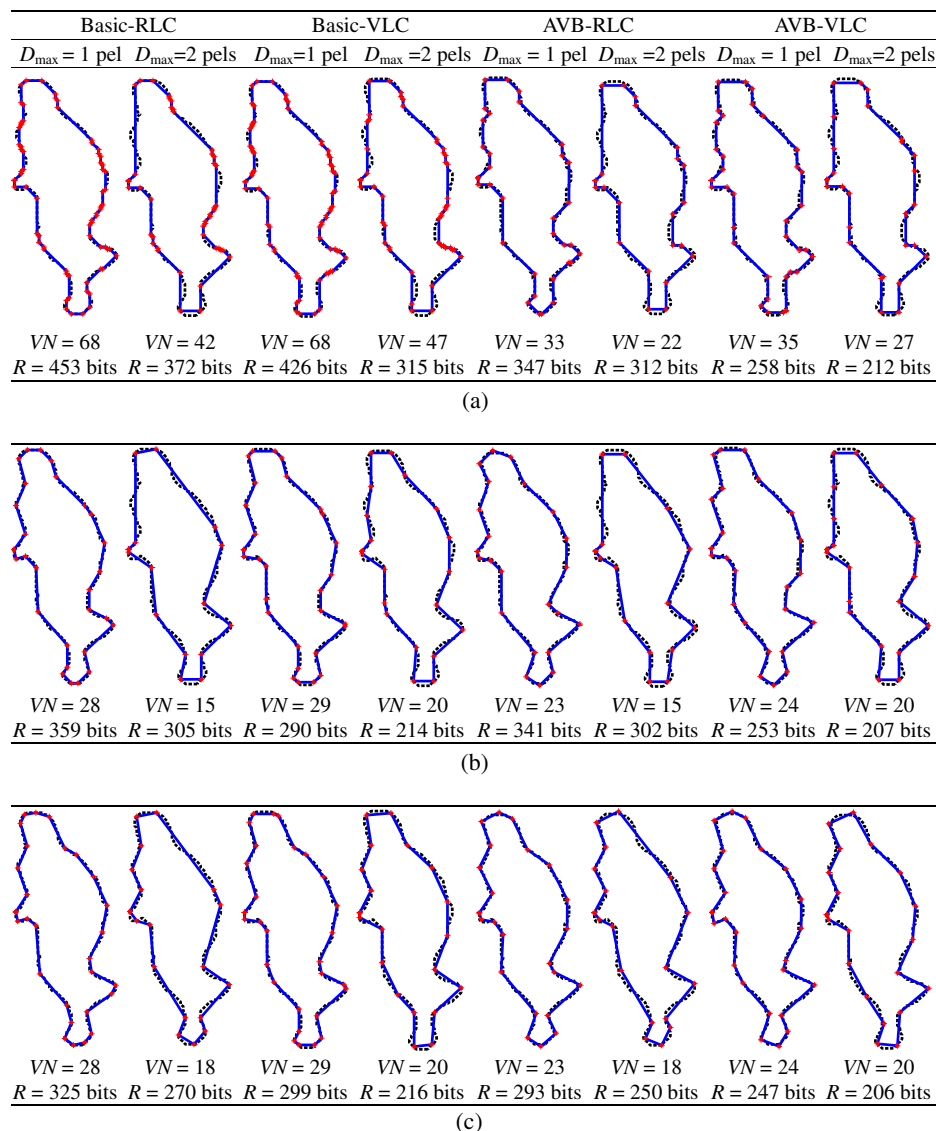
Figure 6 shows the polygons generated with various ORD optimal algorithms. Considering the first column as an example, we can see that the traditional eight-direction structure needs 68 vertices for contour approximation, but both of our structures only need 28 vertices. As a result, the traditional one needs 453 bits, whereas our two proposals need 359 and 325 bits for contour encoding, resulting in 94 and 128 bit savings, respectively.

Moreover, we see that the polygons approximated by our proposals are quite compact and have strong ability to reflect the characteristics of the original object contour, since a relatively straight contour segment is easier to be approximated by an edge, so that the turns of the polygons are more likely to be the corners of the object contour. This will benefit the successive object-oriented applications. In the next section, we will apply these polygons to hand gesture recognition.

#### 4.2 Arbitrary Direction Edge Encoding Structures for Hand Gesture Recognition

Hand gesture recognition is of great importance in human-computer interaction. Usually, the shape feature is sufficient for successful recognition.<sup>23</sup> However, due to the nature of optical camera, the quality of captured images is sensitive to the lighting conditions and cluttered backgrounds, which makes it very difficult to obtain hand shapes.<sup>24</sup>

Due to the new advent of Kinect depth camera (Shenzhen, China),<sup>25</sup> new opportunities for shape-based hand gesture recognition emerge. With the Kinect depth camera, we collect three hand gesture categories, namely Rock, Paper, and Scissors; each category has 50 samples, which contain various rotations, scales, viewpoints, and even self-occlusions. Then, by thresholding from the nearest depth position with a certain gap, we can obtain the hand shapes robustly. Some exemplars, including their color maps, depth maps, and segmented hand shapes, are illustrated in Fig. 7.



**Fig. 6** Description ability comparisons among various ORD optimal algorithms on the Stefan object (Legends—solid line: approximating polygon; dashed line: original contour; asterisk: vertex). (a) Traditional eight-direction edge encoding structure, (b) proposed eight-sector edge encoding structure, (c) proposed 16-sector edge encoding structure.



**Fig. 7** Illustration of our hand gesture dataset captured by the Kinect camera. The first and second columns are the color and depth maps in our dataset, and the third column is the segmentations of hands.

However, even with the help of Kinect camera, due to the limited camera resolution and the complex monitoring environment, quantization error and contour noise are almost inevitable in shape acquisition. Since the ORD optimal algorithms with our two arbitrary edge encoding structures can approximate the contour segments with arbitrary directions, quantization error, and contour noise, it is suitable to handle this recognition task.

Here, we only choose RLC to guarantee that there is no limitation on the encoding edge length. We approximate the hand shape contour by the polygons using our proposals. It is important to determine a proper  $D_{\max}$ . We let  $D_{\max}$  be proportional to the radius of the maximal inscribed disk of the hand shape, so that  $D_{\max}$  is adaptive to the size of palm. To be specific, we set  $D_{\max}$  to 30% of the radius, as this can well discriminate nonsalient fingers from large shape deformations. Only the negative turns of the polygons are considered, since it was reported that the negative corners are much more informative than the positive ones.<sup>26</sup> Suppose

that there are  $k$  negative turns, we classify a hand gesture to Rock if  $k = 0$ , Paper if  $k \geq 3$ , and Scissors otherwise.

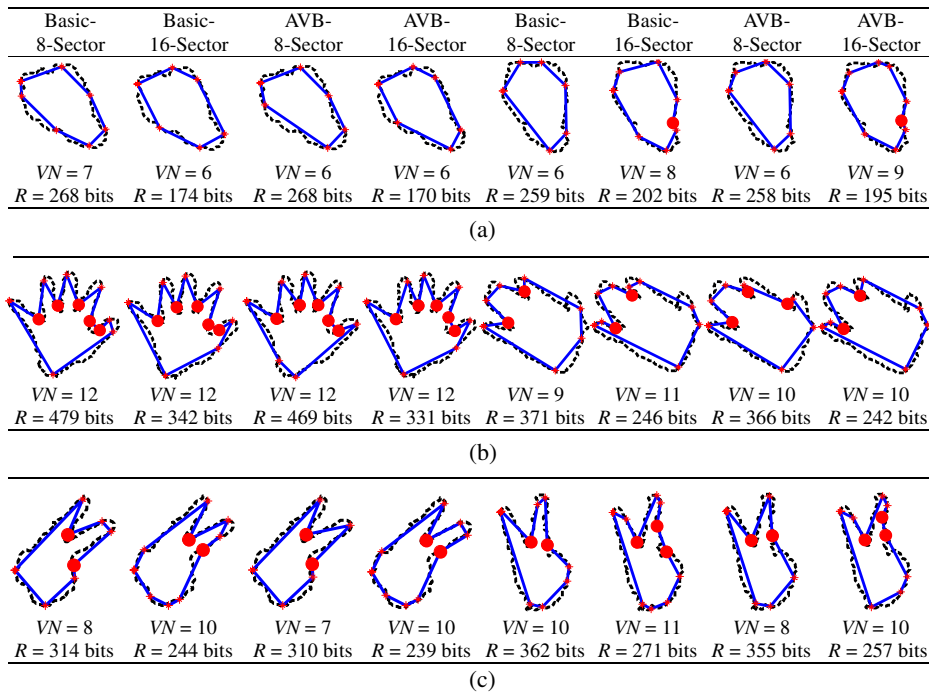
Table 2 provides the confusion matrices and the mean recognition accuracies for various ORD optimal algorithms with our two arbitrary direction structures. As expected, the mean accuracies of the eight-sector structure are much higher than those of the 16-sector structure due to the direction bias effect as mentioned in Sec. 3.2.

To further reveal this direction bias effect, Fig. 8 shows some results selected from our collected dataset. The odd columns show the hand gestures correctly recognized by both of our structures, while the even columns show those correctly recognized by the eight-sector structure but wrongly recognized by the 16-sector structure, except the Paper results which are wrongly recognized by the Basic-16-Sector algorithms. We take the second Rock gesture recognized by the Basic algorithms as an example. Although the 16-sector structure can reduce the total rates from 259 to 202 bits compared with the eight-sector

**Table 2** Confusion matrices for Basic-8-Sector/Basic-16-Sector and AVB-8-Sector/AVB-16-Sector algorithms. The mean recognition accuracies are 93.33%/86% and 94%/76%, respectively. For each gesture category, the higher recognition accuracy between the 8-sector structure and the 16-sector structure is marked in bold. We can see that 8-sector structure always results in higher recognition accuracy, endorsing the earlier comment about 8-sector structure having stronger description ability than 16-sector one.

	<i>Rock</i>	<i>Paper</i>	<i>Scissors</i>
<i>Rock</i>	<b>0.94</b> /0.86, <b>0.94</b> /0.8	0/0, 0/0	0.06/0.14, 0.06/0.2
<i>Paper</i>	0/0, 0/0	<b>0.94</b> / <b>0.94</b> , <b>0.96</b> /0.94	0.06/0.06, 0.04/0.06
<i>Scissors</i>	0.04/0.02, 0.04/0.02	0.04/0.2, 0.04/0.44	<b>0.92</b> /0.78, <b>0.92</b> /0.54





**Fig. 8** Hand gesture recognition by the negative turn detection (Legends—solid line: approximating polygon; dashed line: original contour; asterisk: vertex; dot: negative turn). (a) Contour approximations of the Rock gestures, (b) contour approximations of the Paper gestures, (c) contour approximations of the Scissors gestures.

structure, the bottom right contour segment of the direction near the restricted direction of zero is approximated by two edges of the directions near the preferential directions of  $\mp \arctan(1/2)$  due to the direction bias effect, and therefore, a false-negative turn is generated. It demonstrates that the 16-sector structure does well in compression but is not good at description for object-oriented applications, which is the opposite of the eight-sector structure.

## 5 Conclusions and Future Work

In this paper, we have presented two edge encoding structures, namely 8- and 16-sector structures, for the ORD optimal framework, to relax the direction restriction of the traditional eight-direction structure. Both consist of the sector number, the short component magnitude, and the long component magnitude, which are encoded with the optimal code words under two probability distribution assumptions. Although these two structures are quite simple, experiments on both shape coding and hand gesture recognition validate that the ORD optimal framework with our proposals can achieve high coding gains and mean recognition rates, which makes our proposals potentially promising in the applications of both data compression and pattern recognition fields.

Several research directions based on our proposals are possible. RLC has no limitation on the encoding edge length but is of low efficiency; thus, it is applicable to the object-oriented applications but not to the storage and transmission, whereas VLC exactly opposes it. How to combine their advantages and avoid disadvantages is one research direction, with some preliminary results shown in Ref. 27. Moreover, although our arbitrary direction structures avoid redundant short edges, they still cause noticeable edge errors.

How to reduce them, so that the approximate polygon can be applicable to the more challenging shape-based depth map coding<sup>28,29</sup> and shape-based vision tasks,<sup>30,31</sup> is another research direction, with some preliminary results shown in Ref. 32.

## Acknowledgments

The authors would like to thank Hans Georg Musmann, Michael Höttera, and Jörn Ostermann for providing the MPEG-4 binary shape sequences on the Internet, and Shaojun Zhu for collecting the hand gestures with a Kinect camera. The authors wish to thank Baolan Yan, Zhe Wang, and Zhou Ren for providing their source codes. The authors also want to thank Junsong Yuan, Wenyu Liu, and Chun Wang for their discussions, and two anonymous reviewers for their suggestions. This work was supported in part by the National Natural Science Foundations of China (Grant Nos. 61303238 and 61202460), the Hubei Provincial Natural Science Foundation of China (Grant No. 2013CFB214), the Wuhan Municipal Key Technologies Research and Development Programs (Grant Nos. 2013011001010484, 2013011001010463, and 201250499145-24), and the Wuhan Innovative Talent Development Fund to Dr. Tonghui Qian.

## References

1. A. K. Katsaggelos et al., "MPEG-4 and rate-distortion-based shape-coding techniques," *Proc. IEEE* **86**(6), 1126–1154 (1998).
2. K. J. O'Connell, "Object-adaptive vertex-based shape coding method," *IEEE Trans. Circuits Syst. Video Technol.* **7**(1), 252–255 (1997).
3. J. Chung et al., "A new vertex-based binary shape coder for coding efficiency," *Signal Process.: Image Commun.* **15**(7–8), 665–684 (2000).
4. C.-M. Kuo, C.-H. Hsieh, and Y.-R. Huang, "A new adaptive vertex-based binary shape coding technique," *Image Vis. Comput.* **25**(6), 863–872 (2007).

5. T. Adamek and N. E. O'Connor, "A multiscale representation method for nonrigid shapes with a single closed contour," *IEEE Trans. Circuits Syst. Video Technol.* **14**(5), 742–753 (2004).
6. E. Attalla and P. Siy, "Robust shape similarity retrieval based on contour segmentation polygonal multiresolution and elastic matching," *Pattern Recogn.* **38**(12), 2229–2241 (2005).
7. C. Xu, J. Liu, and X. Tang, "2D shape matching by contour flexibility," *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(1), 180–186 (2009).
8. G. M. Schuster and A. K. Katsaggelos, "An optimal polygonal boundary encoding scheme in the rate distortion sense," *IEEE Trans. Image Process.* **7**(1), 13–26 (1998).
9. G. M. Schuster, G. Melnikov, and A. K. Katsaggelos, "Operationally optimal vertex-based shape coding," *IEEE Signal Process. Mag.* **15**(6), 91–108 (1998).
10. F. A. Sohel, L. S. Dooley, and G. C. Karmakar, "New dynamic enhancements to the vertex-based rate-distortion optimal shape coding algorithm," *IEEE Trans. Circuits Syst. Video Technol.* **17**(10), 1408–1413 (2007).
11. F. A. Sohel et al., "Sliding-window designs for vertex-based shape coding," *IEEE Trans. Multimedia* **14**(3), 683–692 (2012).
12. L. P. Kondi, G. Melnikov, and A. K. Katsaggelos, "Joint optimal object shape estimation and encoding," *IEEE Trans. Circuits Syst. Video Technol.* **14**(4), 528–533 (2004).
13. F. A. Sohel, L. S. Dooley, and G. C. Karmakar, "Accurate distortion measurement for generic shape coding," *Pattern Recogn. Lett.* **27**(2), 133–142 (2006).
14. F. A. Sohel, L. S. Dooley, and G. C. Karmakar, "Fast distortion measurement using chord-length parameterization within the vertex-based rate-distortion optimal shape coding framework," *IEEE Signal Process. Lett.* **14**(2), 121–124 (2007).
15. Z. Lai et al., "Perceptual distortion measure for polygon-based shape coding," *IEICE Trans. Inf. Syst.* **E96.D**(3), 750–753 (2013).
16. F. A. Sohel et al., "Geometric distortion measurement for shape coding: a contemporary review," *ACM Comput. Surv.* **43**(4), 1–29 (2011).
17. G. Melnikov, G. M. Schuster, and A. K. Katsaggelos, "Simultaneous optimal boundary encoding and variable-length code selection," in *Proc. IEEE Int. Conf. on Image Processing*, pp. 256–260, IEEE Computer Society, Los Alamitos, California (1998).
18. G. Melnikov, G. M. Schuster, and A. K. Katsaggelos, "Shape coding using temporal correlation and joint VLC optimization," *IEEE Trans. Circuits Syst. Video Technol.* **10**(5), 744–754 (2000).
19. G. Melnikov and A. K. Katsaggelos, "Shape approximation through recursive scalable layer generation," in *Proc. IEEE Int. Conf. on Image Processing*, pp. 915–918, IEEE Inc., Piscataway, New Jersey (2000).
20. G. M. Schuster and A. K. Katsaggelos, *Rate-Distortion Based Video Compression: Optimal Video Frame Compression and Object Boundary Encoding*, Kluwer, Norwell, MA (1997).
21. T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms*, McGraw-Hill, New York (1991).
22. J. De Winter and J. Wagemans, "The awakening of atneave's sleeping cat: identification of everyday objects on the basis of straight-line versions of outlines," *Perception* **37**(2), 245–270 (2008).
23. J. P. Wachs et al., "Vision-based hand-gesture applications," *Commun. ACM* **54**(2), 60–71 (2011).
24. A. Erol et al., "Vision-based hand pose estimation: a review," *Comput. Vis. Image Und.* **108**(1–2), 52C73 (2007).
25. Z. Zhang, "Microsoft Kinect sensor and its effect," *IEEE Multimedia* **19**(2), 4–10 (2012).
26. J. De Winter and J. Wagemans, "Perceptual saliency of points along the contour of everyday objects: a large-scale study," *Percept. Psychophys.* **70**(1), 50–64 (2008).
27. J. Zhu et al., "Adaptive edge encoding schemes for the rate-distortion optimal polygon-based shape coding," in *Proc. IEEE Data Compression Conf.*, pp. 103–112, IEEE Computer Society, Los Alamitos, California (2014).
28. N. Zhang, S. Ma, and W. Gao, "Shape-based depth map coding," in *Proc. Fifth Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 316–319 (2009).
29. K.-J. Oh, A. Vetro, and Y.-S. Ho, "Depth coding using a boundary reconstruction filter for 3-D video systems," *IEEE Trans. Circuits Syst. Video Technol.* **21**(3), 350–359 (2011).
30. C. Wang et al., "Perceptually friendly shape decomposition by resolving segmentation points with minimum cost," *J. Vis. Commun. Image Represent.* **24**(3), 270–282 (2013).
31. C. Li, M. Ovsjanikov, and F. Chazal, "Persistence-based structural recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2003–2010, IEEE Computer Society, Los Alamitos, California (2014).
32. Z. Lai, F. Zhang, and W. Lin, "Operational rate-distortion shape coding with dual error regularization," in *Proc. IEEE Int. Conf. on Image Processing*, IEEE Inc., Piscataway, New Jersey (2014).

**Zhongyuan Lai** is an assistant researcher at Jiangnan University. He received his BEng degree in communication engineering from Huazhong University of Science and Technology in 2007 and his PhD degree in communication and information system from the same university in 2013. His current research interests include image compression and pattern recognition. He is a member of IEEE, ACM, and IEICE.

**Junhuan Zhu** is a PhD student at the University of Rochester. He received his BEng degree in electronics and information engineering from Huazhong University of Science and Technology in 2010, and his MS degree in electrical and computer engineering from the University of Rochester in 2013. His current research interests include image compression and social media data mining.

**Jiebo Luo** joined the University of Rochester in fall 2011 after over 15 years at Kodak Research Laboratories as a senior principal scientist. He has been involved in numerous technical conferences. He is the editor-in-chief of the *Journal of Multimedia* and has served on the editorial boards of numerous international journals. He has authored over 200 technical papers and 70 U.S. patents. He is a fellow of SPIE, IEEE, and IAPR.